





Mémoire Virtuelle

Plan

- Introduction
- Allocation non-contigüe
 - ▣ Segmentation
 - ▣ Pagination
- Gestion de la mémoire virtuelle
 - ▣ Algorithmes de remplacement

Introduction

- La mémoire virtuelle est une fonctionnalité permettant d'exécuter un processus dont la taille de son espace d'adressage excède celle de la mémoire physique (réelle).
- L'espace de mémoire virtuelle est donc indépendant de l'espace de mémoire physique
- Avec la mémoire virtuelle, les programmes ne sont plus contraints par les limites de mémoire physique.

Introduction

La mémoire virtuelle est rendu possible grâce à deux concepts fondamentaux :

- ▣ Segmentation
- ▣ Pagination
- ▣ Ces deux concepts ont vu le jour pour remédier principalement au problème de la fragmentation
 - ▣ Ils permettent une allocation de mémoire non-contiguë

Introduction : Allocation non-contiguë

- L'allocation non-contiguë consiste à stocker l'espace d'adressage d'un processus d'une façon dispersée dans la mémoire physique
- Diviser un programme en morceaux plus petits
- Allouer à chaque morceau un espace physique séparé
 - Utilisation efficace des petits trous
 - Diminution importante de la fragmentation externe

Plan

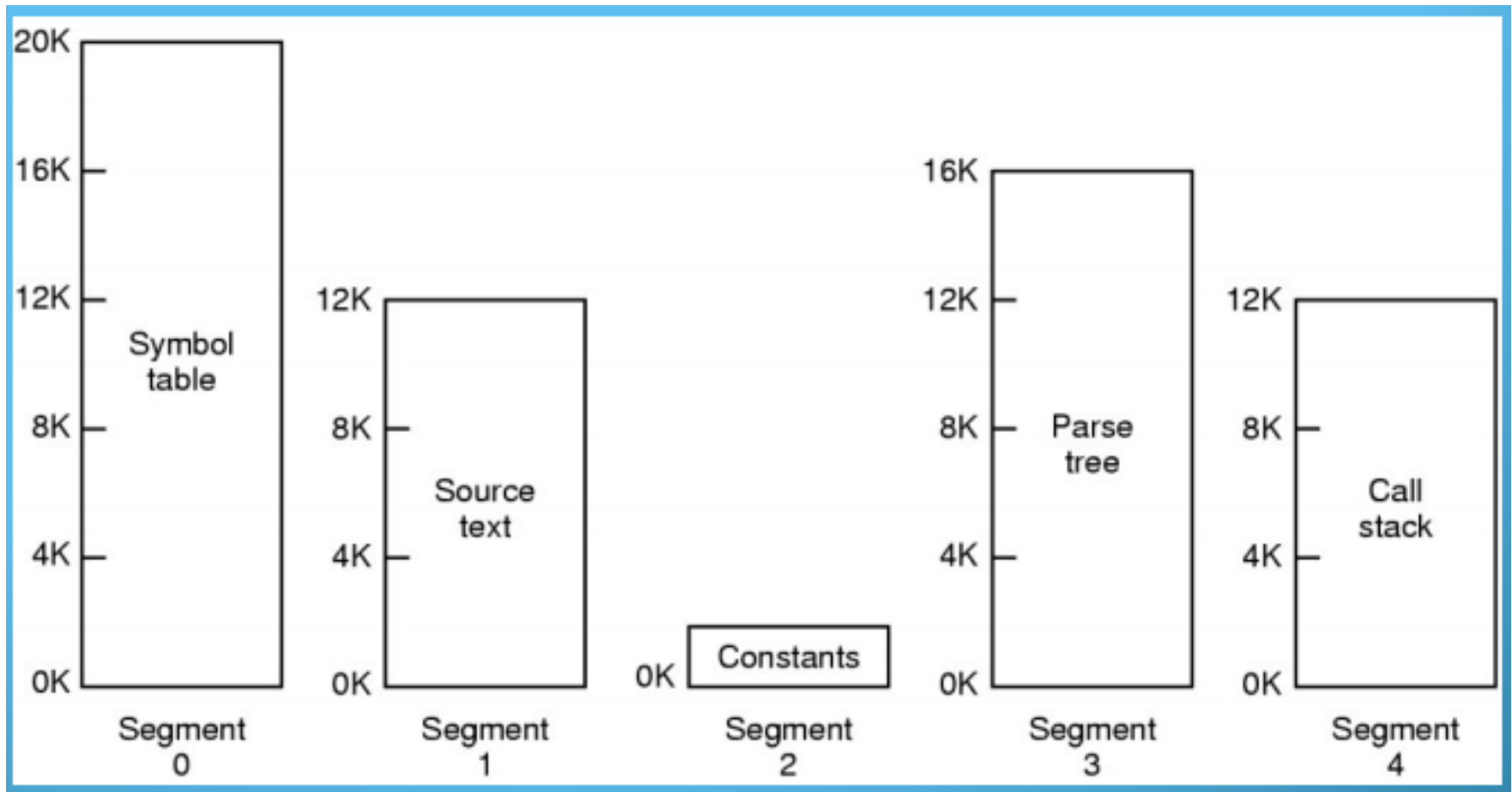


- Introduction
- Allocation non-contigüe
 - ▣ Segmentation
 - ▣ Pagination
- Gestion de la mémoire virtuelle
 - ▣ Algorithmes de remplacement

Segmentation

- ❑ Consiste à diviser d'une façon logique (des modules) l'espace d'adressage d'un processus en plusieurs segments.
- ❑ L'espace d'adressage logique correspond alors à un ensemble de segments de tailles variables et indépendants.
- ❑ Les segments peuvent croître ou diminuer dynamiquement et indépendamment des autres segments.
- ❑ Au moins l'espace d'adressage logique est divisé en trois segments.

Segmentation

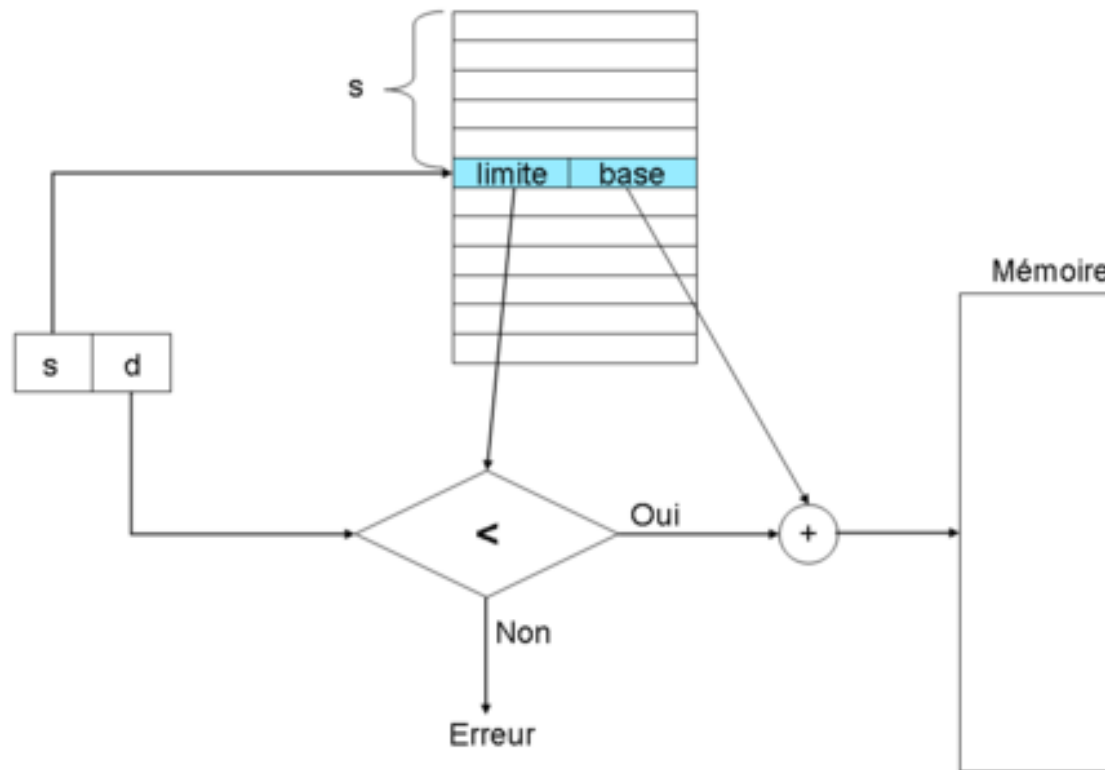


Segmentation

- Chaque segment a un numéro et une longueur.
- Chaque adresse relative (virtuelle) est définie par le **numéro du segment**, auquel appartient l'information, et son **décalage** par rapport au segment.
- Les segments d'un processus sont référencier dans une table contenant les champs suivants :
 - ▣ Numéro du segment
 - ▣ Registre base
 - ▣ Registre limite ou longueur
- Les informations stockées dans la table des segment servent aussi un outil de protection des frames accordés à ces segment.

Segmentation

- L'adresse virtuelle est représenté par un couple composé du N°segment, déplacement est traduite en **adresse physique** par le biais de la table de segments associé au processus.
- Un test est effectué pour vérifier que le décalage est bien dans l'intervalle du segment.



Segmentation

- La segmentation a permis l'idée que pas forcément l'intégralité du programme soit présente dans la mémoire principale au moment d'exécution.
- Mais les segments d'un processus sont de tailles variables, ce qui pose encore le problème de la fragmentation externe
- Solution : La pagination
 - ▣ des unités d'allocation mémoire de tailles égales

Plan



- Introduction
- Allocation non-contigüe
 - ▣ Segmentation
 - ▣ **Pagination**
- Gestion de la mémoire virtuelle
 - ▣ Algorithmes de remplacement

Pagination

- La pagination consiste à répartir le processus en sous parties de taille égales, qu'on appelle **pages**.
- La mémoire physique est divisée en blocs de même taille, qu'on appelle cadres ou **frames**.
- La taille des cadres est généralement une puissance de 2 (entre 512 et 8192 octets).
- La taille d'une page de la mémoire virtuelle est égale à la taille d'un cadre de la mémoire physique.

Pagination

Les pages d'un processus peuvent donc être assignées aux cadres disponibles n'importe où en mémoire principale.

- Un processus peut être éparpillé dans la mémoire physique.
- La fragmentation externe est éliminée.
- Seule la dernière page d'un processus peut souffrir de la fragmentation interne.

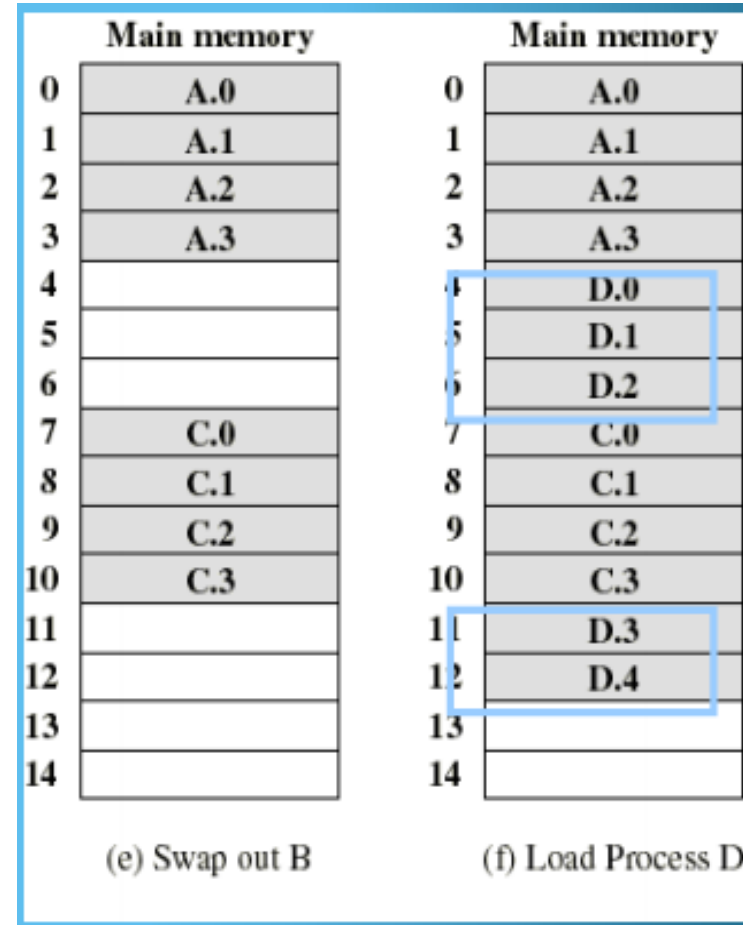
Pagination

La pagination impose que le système :

- Ait en permanence des informations sur les frames libres et occupés de la mémoire physique.
- Trouve **n** frames libres si un programme en a besoin.
- Puisse gérer la correspondance entre adresses physiques et logiques.

Pagination : exemple

- Supposons que le processus B se termine ou est suspendu.
- Nous pouvons maintenant transférer en mémoire un processus D, qui demande 5 cadres, bien qu'il n'y ait pas 5 cadres contigus disponibles.



Pagination : Adresse virtuelle

- L'adresse virtuelle, dans la pagination, est composé du **numéro de page**, auquel appartient l'information, et d'un **déplacement** relatif au début de la page.
- L'adresse virtuelle est codée sur 16 bits.
 - ▣ Les 4 bits de poids fort indiquent le numéro de page.
 - ▣ Les autres bits donnent le décalage par rapport au début de la page (déplacement dans la page).

Pagination : Adresse virtuelle

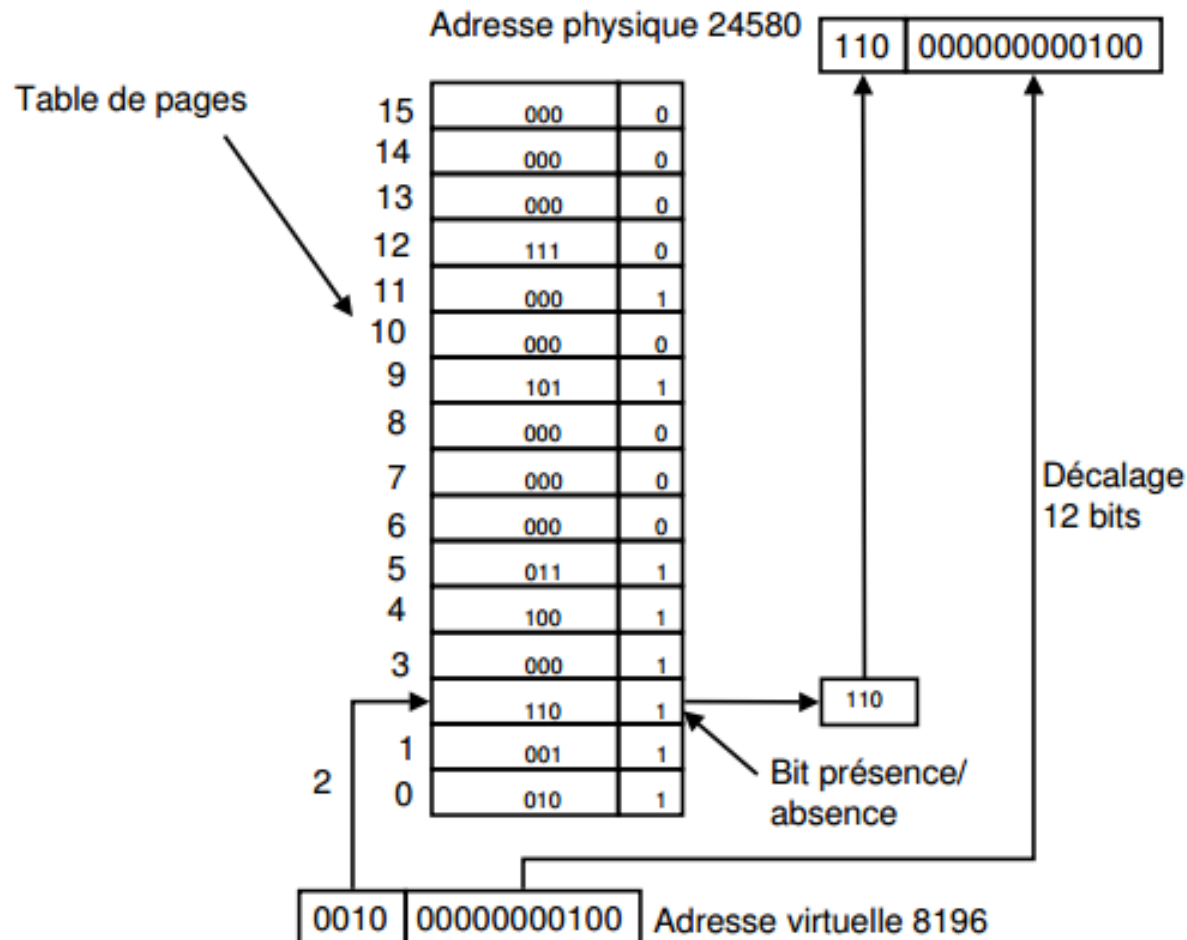
- A chaque processus est associée une Table de pages mémorisant principalement la correspondance entre le **n° de la page** (virtuelle) et le **n° du frame contenant cette page**.
- Chaque entrée de la Table de pages est composée de plusieurs champs :
 - ▣ Bit de présence.
 - ▣ Bit de référence (R).
 - ▣ Bits de protection.
 - ▣ Bit de modification (M).
 - ▣ **Le numéro de cadre.**
- Le nombre d'entrées dans cette table est égal au nombre de pages virtuelles composant l'espace d'adressage du processus.

Pagination : Adresse physique

Le MMU examine l'entrée dans la **table de pages** qui correspond au numéro de page.

- Il détermine l'adresse physique :
 - ▣ en recopiant dans les 3 bits de poids le plus fort le numéro de cadre correspondant au numéro de page,
 - ▣ et dans les 12 bits de poids le plus faible les bits du décalage de l'adresse virtuelle.
- Les deux parties constituant l'adresse physique

Pagination : Conversion d'adresses



Pagination : conversion d'adresses

- Dans la table de pages, le bit de validité permet au système de vérifier la présence d'une page en mémoire physique.
 - ▣ 1 : page est en mémoire, 0 : page n'est pas en mémoire
 - ▣ Initialement tous les bits sont mis à 0.
 - ▣ Au moment de la conversion des adresses, si le bit est à 0 ; défaut de page (page fault).
 - ▣ Le processeur est donné à un autre processus pendant la gestion de la faute.

Plan



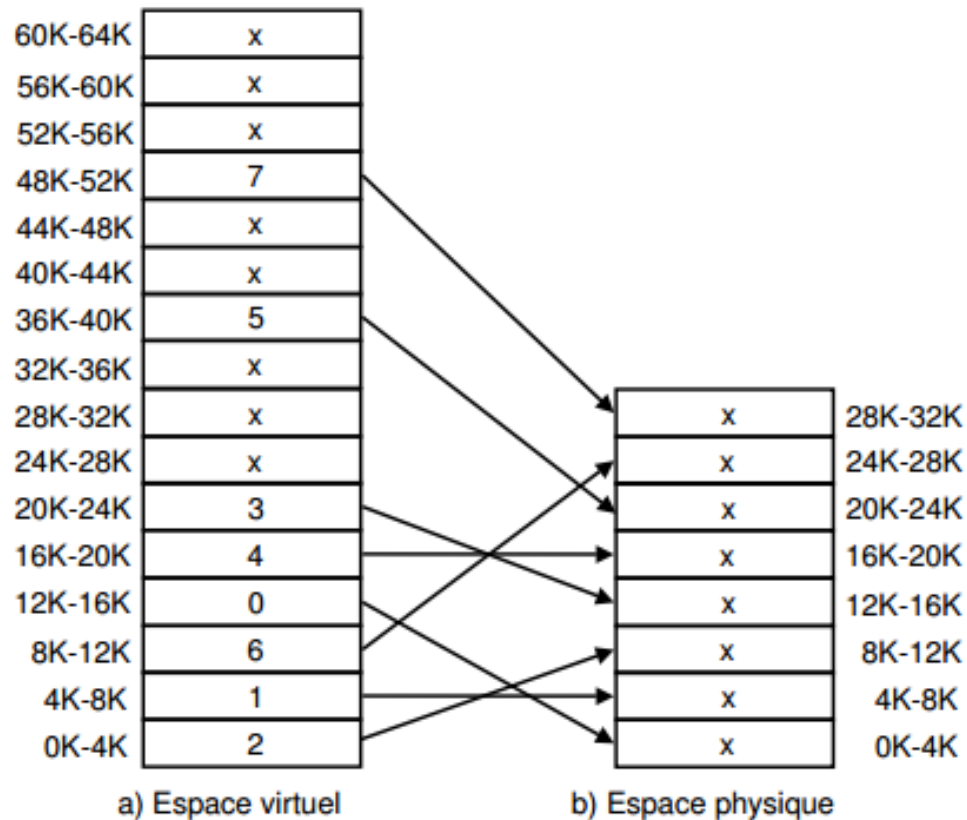
- Introduction
- Allocation non-contigüe
 - ▣ Segmentation
 - ▣ Pagination
- Gestion de la mémoire virtuelle
 - ▣ Algorithmes de remplacement

Mémoire virtuelle

La séparation de la mémoire logique et la mémoire physique implique :

- Seuls de petites parties des programmes ont besoin d'être en mémoire pour l'exécution.
- L'exécution peut continuer à condition que la prochaine instruction (ou donnée) soit dans une page se trouvant en mémoire principale.
- Les pages du processus peuvent être déplacées à différentes régions de la mémoire.
- Une image de tout l'espace d'adressage du processus est gardée en mémoire secondaire.
- Les pages manquantes pourront être prises au besoin par le swapping.

Mémoire virtuelle



Mémoire virtuelle

Le système en plus d'allocation, il doit gérer le remplacement des pages d'une façon optimale :

- ▣ Le remplacement de pages :
 - Trouver une page à sortir de la mémoire : page victime.
 - Plusieurs cadres de mémoire ne peuvent pas être `victimes` :
p.ex. cadres contenant le noyau du SE
 - La page *victime* doit être réécrite en mémoire secondaire si elle a été modifiée, sinon, sa copie sur disque est encore fidèle.

Plan



- Introduction
- Allocation non-contigüe
 - ▣ Segmentation
 - ▣ Pagination
- Gestion de la mémoire virtuelle
 - ▣ Algorithmes de remplacement

Algorithmes de remplacement de pages

- Objectif : assurer l'exécution avec le plus petit nombre de remplacement possible.
 - ▣ Choisir la victime de façon à minimiser le taux de défaut de pages.
- Plusieurs méthodes pour choisir qui sort de la mémoire :
 - ▣ **First-in-First-Out**
 - ▣ **Not Recently Used**
 - ▣ **Least Recently Used**
 - ▣ **Deuxième chance : horloge (clock)**
 - ▣ **Algorithmes basés sur compteurs**
 - ▣ ...

FIFO : First-in-First-Out

- Quand une page doit être remplacée, on choisit la plus vieille.
 - Pour chaque page on connaît son heure d'arrivée en mémoire.
- Problème :
 - Les premières pages amenées en mémoire sont souvent utiles pendant toute l'exécution d'un processus
 - variables globales, programme principal, etc.

NRU : Not Recently Used

- Consiste à remplacer la page qui n'a pas été utilisée depuis longtemps;
- Il utilise les bits **R** et **M** (**table de page**) associés à chaque page pour déterminer les pages non récemment utilisées.
- Le bit **R** est positionné à **1** chaque fois qu'une page est référencée (utilisée). Il est remis à **0** à chaque interruption d'horloge.
- Le bit **M** est positionné à **1** lorsque la page est modifiée
 - ▣ elle n'est plus identique à sa copie sur disque

NRU : Not Recently Used

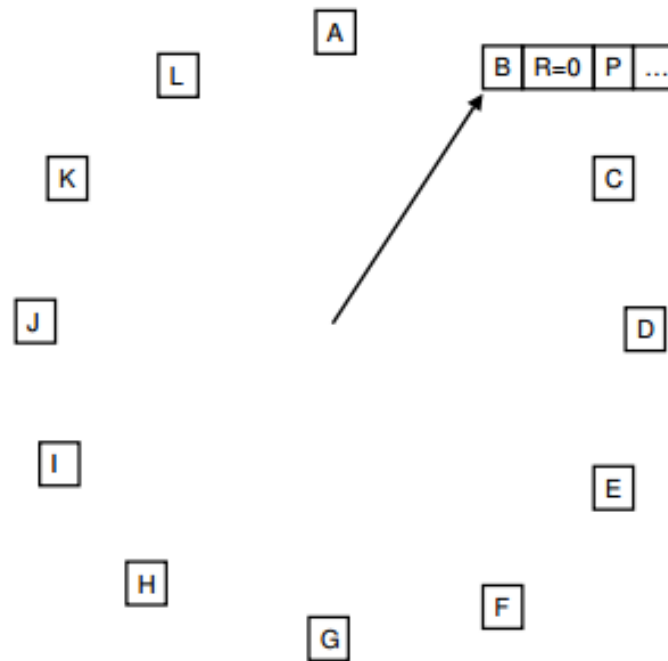
- Lorsqu'un défaut de page se produit, l'algorithme NRU sélectionne la page à retirer selon les priorités suivantes :
 1. Page non référencée et non modifiée (**R=0, M=0**).
 2. Page non référencée et modifiée (**R=0, M=1**).
 3. Page référencée et non modifiée (**R=1, M=0**).
 4. Page référencée et modifiée (**R=1, M=1**).
- ▣ Dans tous les cas si la page victime est modifiée (**M=1**) alors on doit la sauvegarder sur le disque avant de l'expulser.

LRU : Least Recently Used

- Consiste à remplacer la page qui n'a pas été utilisée depuis le plus longtemps
- L'algorithme LRU mémorise dans une liste chaînée toutes les pages en mémoire.
- La page la plus utilisée est en tête de liste et la moins utilisée est en queue.
- Lorsqu'un défaut de page se produit, la page la moins utilisée est retirée.

Algorithme de l'horloge

Dans cet algorithme, les pages en mémoire sont mémorisées dans une liste circulaire en forme d'horloge.



Algorithme de l'horloge

- Appelé aussi algorithme de la seconde chance :
 - Les cadres qui viennent d'être utilisés ($R=1$) ne sont pas remplacés, on leur donne une deuxième chance.
- Lorsqu'un défaut de page se produit, la page pointée par l'indicateur est examinée.
 - Si le bit R de la page pointée par l'indicateur est à 0, la page est retirée, la nouvelle page est insérée à sa place et l'indicateur avance d'une position.
 - Le bit R de la page chargée est mis à 1.
 - Sinon, il est remis à 0 et l'indicateur avance d'une position.
 - Cette opération est répétée jusqu'à ce qu'une page ayant R égal à 0 soit trouvée.

Algorithmes basés sur compteurs

Garder un compteur, propre à chaque cadre, incrémenté à chaque utilisation de la page y est chargée

- LFU: Least Frequently Used
 - ▣ remplacer la pages avec le plus petit compteur
- MFU: Most Frequently Used:
 - ▣ remplacer les pages bien utilisées pour donner une chance aux nouvelles